



**Department of Electronics & Communication Engineering  
National Institute of Technology Rourkela  
Rourkela – 769008**

# **IMAGE SEGMENTATION AND 3-D RECONSTRUCTION OF CORONARY ARTERY**

**B.TECH PROJECT THESIS**

**SUBMITTED BY:**

**ARPAN SURAVI PRASAD**

**111EC0183**

**SUPERVISOR:**

**Prof. K. K. MAHAPATRA**

**Department of Electronics & Communication Engineering**

**National Institute of Technology, Rourkela-769008**



## **NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA**

### **CERTIFICATE**

This is to certify that the thesis entitled, “**IMAGE SEGMENTATION AND 3-D RECONSTRUCTION OF CORONARY ARTERY**” submitted by **Arpan Suravi Prasad** bearing Roll No – **111EC0183** in partial fulfilment of the requirements for the award of Bachelor of Technology degree in **Electronics and Communication Engineering** during the session 2014-15 at National Institute of Technology, Rourkela (Deemed University) is an authentic work done by him under my supervision and guidance.

To the best of my knowledge, the matter contained in this thesis has not been submitted to any other university/institute for the award of any Degree/Diploma.

Place: Rourkela

**Prof K. K Mahapatra**

Date:

Dept. of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela-769008

I

## **ACKNOWLEDGEMENT**

I am very grateful to my thesis guide Prof. Kamalakanta Mahapatra for giving me this opportunity to work under him throughout the whole time-span of this Project work. Without his guidance, motivation and invaluable inputs the project would not have been possible.

I would like to thank CYBORG The club for Robotics and Automation for enhancing my technical skills.

I would also like to thank Shikha Singhal and Salakha Singhal for their encouragement and support.

It was a great learning experience in the mentally stimulating ambience of the institute.

ARPAN SURAVI PRASAD

111EC0183

**DEDICATED TO “MY MOTHER”**

## **ABSTRACT**

Segmentation of arterial wall boundaries from biomedical images is an important issue for many applications such as in the study of plaque characteristics, to extract mechanical properties of the arterial wall, its 3-D construction, and the measurements such as wall radius, lumen radius and lumen size. So here we present a solution to segmentation of images of coronary arteries and 3D construction. The structure introduced here is a set of connected vertices. An initial contour model which is defined here is allowed to deform according to an energy minimizing term with minimum number of iterations. The energy associated with the contours depends on curvature of contour, and image features. Using training data in a properly built shape space, we are able to classify media and adventitia walls approximately using a large set of data sets using Support Vector Machines. The 3D construction of artery using point matching of successive frames is also explained here with less complexity. The tests of the presented algorithm on a large set of data depicts the effectiveness of this approach.

## **CONTENTS**

ACKNOWLEDGEMENT.....	3
ABSTRACT.....	5
1. Chapter-1.....	
1.1. Active Contour.....	8
1.2.Conformal Mapping.....	11
1.3.Inference.....	16
2. Chapter-2.....	17
2.1. Gradient Approach.....	18
2.2.Manual Segmentation.....	19
2.3.GLCM Matrix.....	19
2.3.1. OFFSET.....	20
2.4.Feature Extraction.....	21
2.5.SVM.....	22
2.6.Results.....	25
3. Chapter-3.....	27
3.1.Algorithm.....	29
3.2.K-d tree.....	30
3.2.1. Construction.....	30
3.2.2. NN Search.....	32
3.2.3. Complexity.....	33
3.3.Transformations.....	33
3.4.Error Minimizations.....	34
3.5.Result of Point Matching.....	35
CONCLUSION.....	36
REFERENCES.....	37

# **CHAPTER-1**

## **INTIMAL SEGMENTATION**

## 1.1. ACTIVE CONTOUR:

The general idea in snakes or active contour models is to deform a contour, with respect to some of the constraints of the given image, in order to detect objects of interest present in the given image. In general, starting with a contour around an object required to be detected, then the contour deforms toward its normal (interior) and is bound to stop on the boundary of desired object.

Let us define the contour  $\mathbf{U}$  in  $\Theta$ , as the boundary of an open subset of  $\pi$  of  $\Theta$  (i.e.  $\pi \in \Theta$ , and  $\mathbf{U} = \partial\pi$ ). In what follows, inside ( $\mathbf{U}$ ) denotes the region  $\pi$ , and outside ( $\mathbf{U}$ ) denotes the region  $\Theta/\pi$ .

Everything in the earth try to minimise its energy. Our method is the minimization of an energy based-segmentation. Let us explain the most basic idea of this model for a simple case. Assume that an image  $U_0$  is formed by two segments of approximately piecewise-constant intensities, which are of distinct values  $U_0^i$  and  $U_0^o$ . Assume further the case that the object needed to be detected is represented by a region with the value  $U_0^i$ . Let us denote  $i$  boundary of the same by  $\mathbf{U}_0$ . Then we have points interior to the object [(or *inside* ( $\mathbf{U}_0$ ))], and  $U_0^i \approx U_0^o$  points exterior to the object [(or *outside* ( $\mathbf{U}_0$ ))]. Now let us have a look on the following energy term:

$$F_1(\mathbf{U}) + F_2(\mathbf{U}) = \int_{\text{inside } \mathbf{U}} |U_0(x, y) - U_1|^2 dx dy + \int_{\text{outside } \mathbf{U}} |U_0(x, y) - U_2|^2 dx dy$$

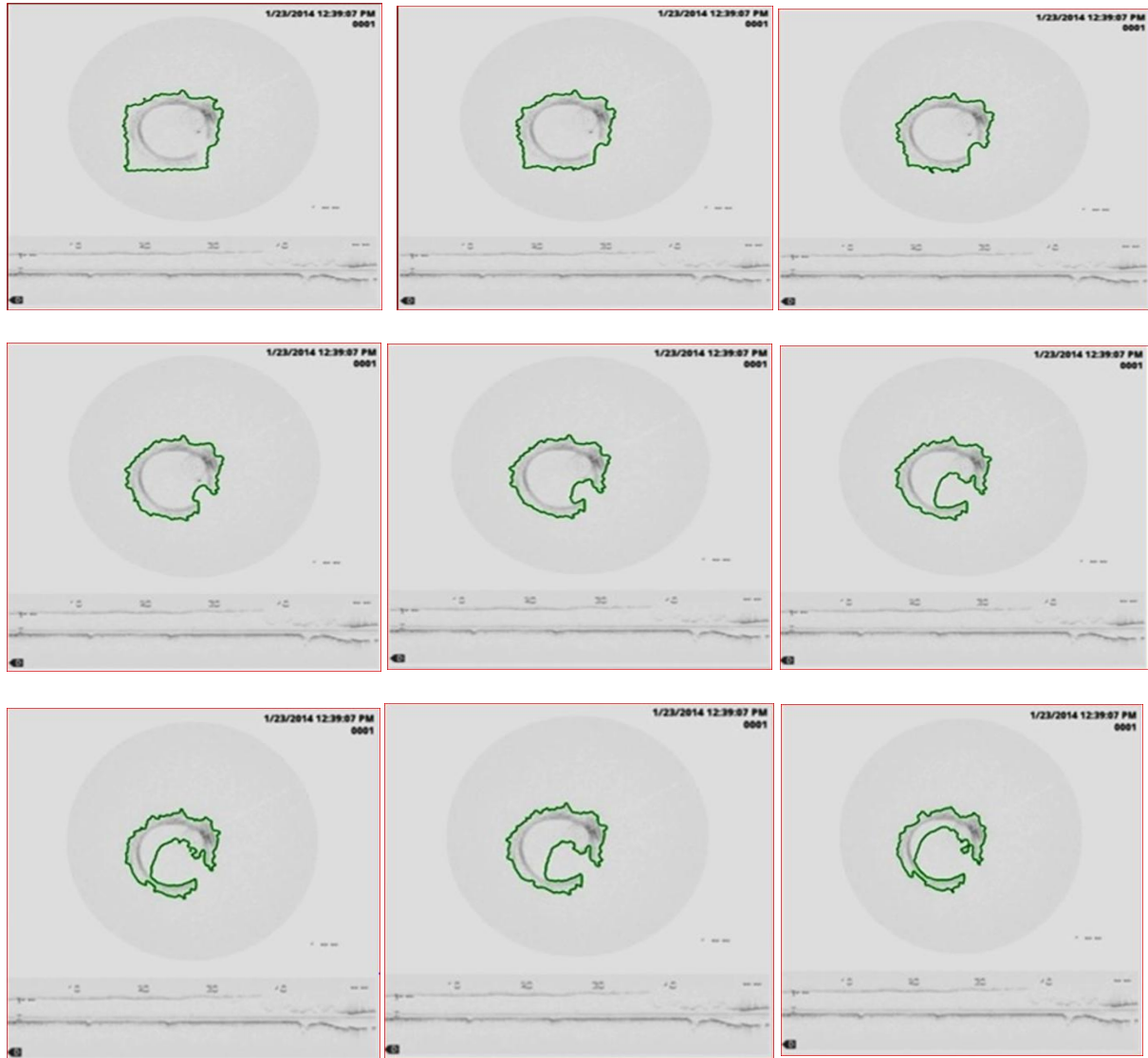
Where  $\mathbf{U}$  is any other variable contour, and the constants  $\mathbf{U}_1, \mathbf{U}_2$ , depending on  $\mathbf{U}$ , are the mean of  $U_0$  inside  $\mathbf{U}$  and outside  $\mathbf{U}$  respectively. In the given case, it is obvious that  $U_0$ , which is the boundary of the object of interest, is the minimum of the energy term.

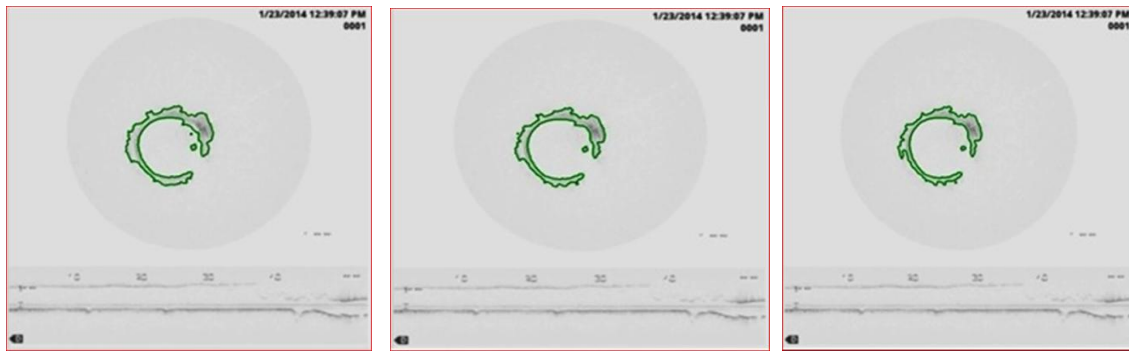
$$\lim_{C \rightarrow \inf} \{F_1(\mathbf{U}) + F_2(\mathbf{U})\} \approx 0 \approx F_1(U_0) + F_2(U_0)$$



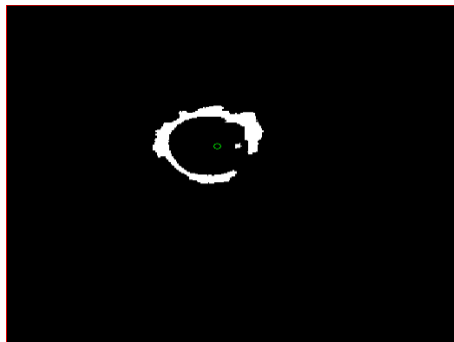
This can be verified easily. If curve  $U$  is outside the object of interest then  $F1(U) > 0$  and  $F2(U) \approx 0$ . Similarly if curve  $U$  is inside the object of interest then  $F2(U) > 0$  and  $F1(U) \approx 0$ . And finally the energy of the contour attains minimum when  $U = U_0$  the condition arises when contour is on the boundary of image.

Applying this Active Contour model to an OCT image the output of the successive iterations can be seen as

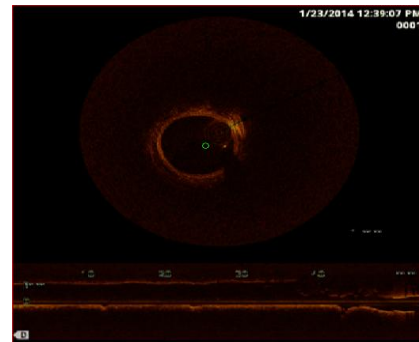




Approximate center of the cross section is calculated as the centroid of the contour.

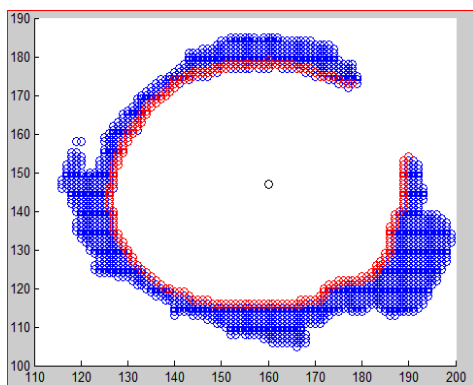


Result after active Contour

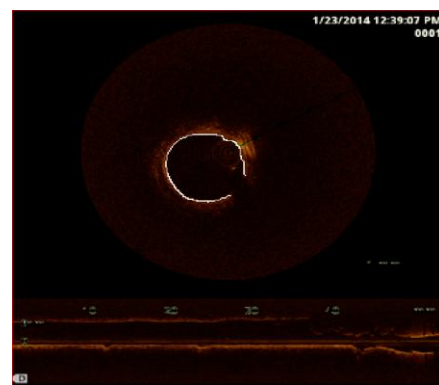


Centre matching on original image

By moving the approximate center of the contour in all direction of the image trace the first white point throughout the image by changing theta and radius to obtain intima. Results obtained is mapped with actual image and **validated for 1500 samples**.



Traced Intima by tracking the interior points



Intimal points mapped to original image

## 1.2. CONFORMAL MAPPING

Taking approximate center('C<sub>0</sub>') of intima as the center of the contour calculated by mean of the co-ordinates of the active contour. Taking 'C<sub>0</sub>' as the center moving 'C<sub>0</sub>' for different values of radius and different angles trace intima as the first point(X<sub>i</sub>,Y<sub>i</sub>) reached after implementing active contour method.

Where X represents the set of x coordinates of intima

$$X=\{X_1, X_2, X_3, \dots, X_N\}$$

And Y represents set of y coordinates of intima.

$$Y=\{Y_1, Y_2, Y_3, \dots, Y_N\};$$

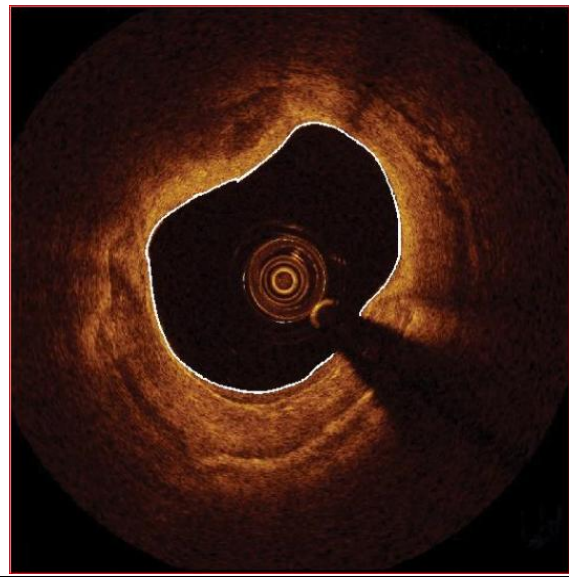
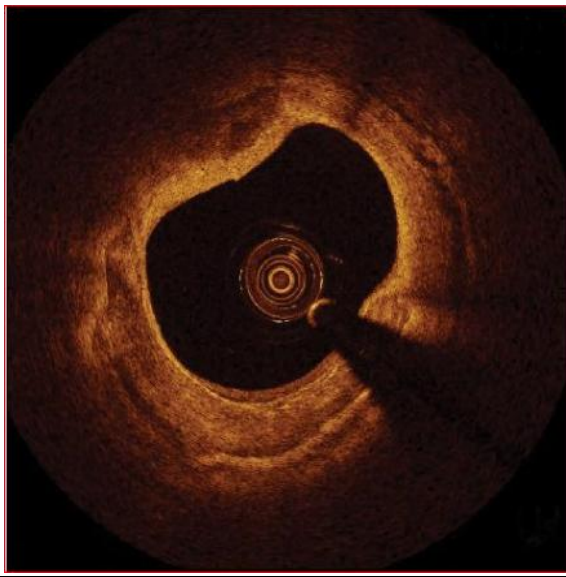
Mapping the intima obtained as a result of active contour onto original image i.e; index

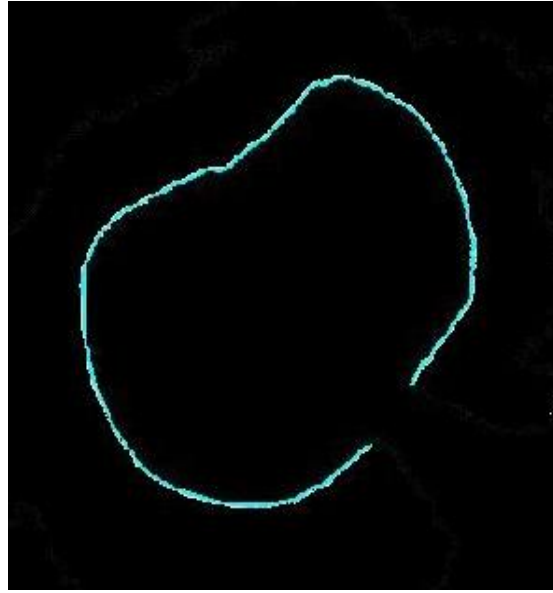
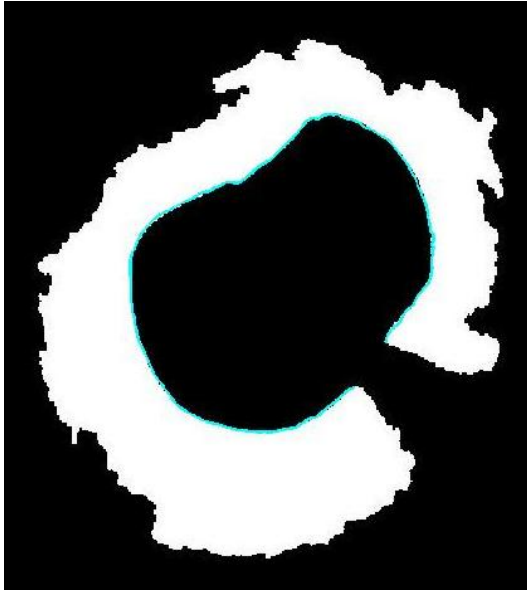
(X<sub>i</sub>,Y<sub>i</sub>) is mapped on to original image I.

Approximate center C<sub>0</sub>(X<sub>0</sub>, Y<sub>0</sub>) of the coronary artery is obtained by approximate center of X and Y

Where

$$X_0 = \frac{\sum_{i=1}^N X_i}{N} \quad \text{and} \quad Y_0 = \frac{\sum_{i=1}^N Y_i}{N}$$





Taking a neighborhood of  $(X_0, Y_0)$  in XY-plane where the neighborhood is represented by

$$X_{cn} = \{X_{c1}, X_{c2}, X_{c3}, \dots, X_{cM}\}$$

$$Y_{cn} = \{Y_{c1}, Y_{c2}, Y_{c3}, \dots, Y_{cM}\}$$

Taking each  $(X_{0i}, Y_{0i})$  as the center mean intimal distance is calculated represented by  $D_{mean}$ .

$$D_{mean} = \{D_{c1}, D_{c2}, D_{c3}, \dots, D_{cM}\}$$

$$\text{Where } D_{ci} = \frac{\sum_{j=1}^N \left( \sqrt{(X_j - X_{ci})^2 + (Y_j - Y_{ci})^2} \right)}{N}$$

$C_1(cc)$  is chosen such that distance from all the  $(X_{ci}, Y_{ci})$  is same to assume intima ideally to circular.

So deviation of the distance from  $(X_{ci}, Y_{ci})$  to  $D_{ci}$  be minimum for  $C_1$ .

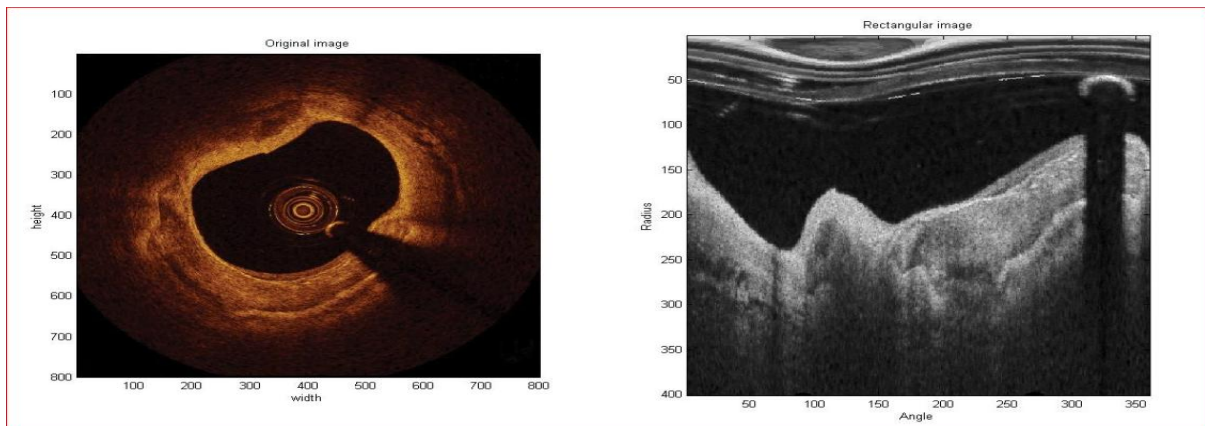
$$C_I(Xc, Yc) = \left\{ \left( (X_{ci}, Y_{ci}) \left| \arg \min_{i=1}^m \left( \sum_{j=1}^N \left( \left( \sqrt{(X_{ci} - X_j)^2 + (Y_{ci} - Y_j)^2} \right) - D_{ci} \right)^2 \right) \right) \right\}$$

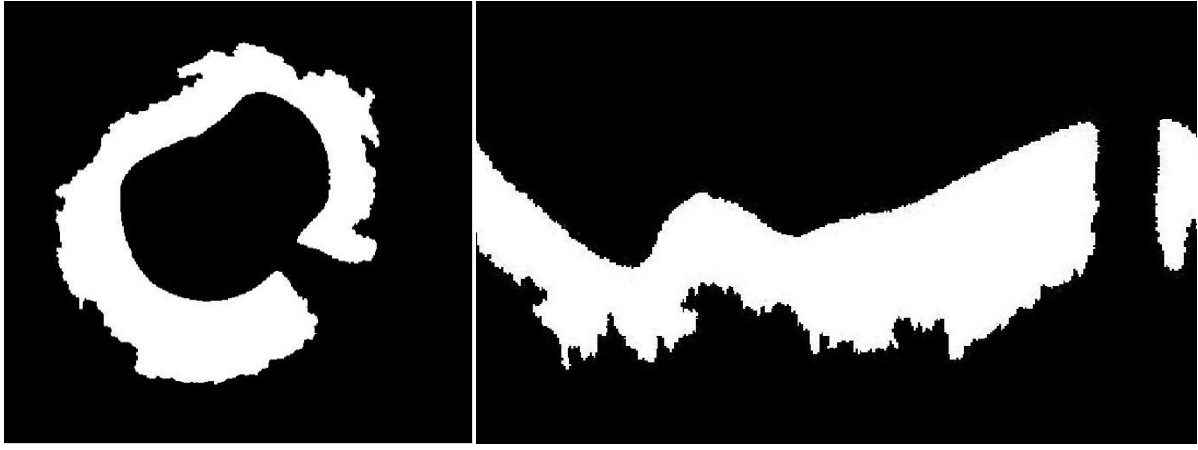
- Taking center as  $C_1(X_c, Y_c)$  mapping the image from (X ,Y) plane to (U , V) plane defined by

Where

$$U_i = \sqrt{(Xi^2) + (Yi^2)}$$

$$V_i = \arctan\left(\frac{Yi}{Xi}\right)$$





Output of Active Contour

Planar Image after Conformal Mapping

Point  $C_1(X_c, Y_c)$  present in XY-plane is mapped to a line  $l$  present in UV-plane represented by  $U=0$ .

Mean line (*mean*) of **Fig.** is obtained by shifting line  $l$  by distance  $D_c$ . So *mean* line is represented by  $U=D_c$ .

Larger the deviation of intima layer from *mean* line larger the deviation of the inner layer from circular. Intima of UV-plane can be represented as

$$U_i = \{U_{i1}, U_{i2}, U_{i3}, \dots, U_{iP}\}$$

$$V_i = \{V_{i1}, V_{i2}, V_{i3}, \dots, V_{iP}\}$$

Deviation ( $\sigma_{\text{intima}}$ ) of intima from *mean* line can be obtained as

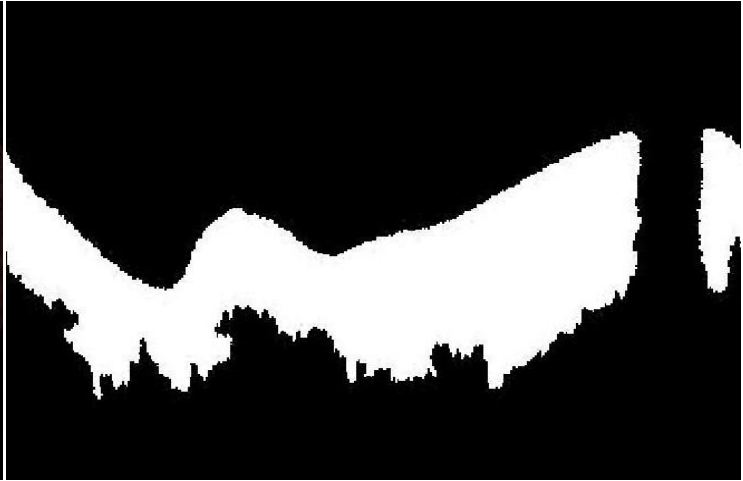
$$\sigma_{\text{intima}} = \sum_{j=1}^p (X_j - D_c)^2$$

As number of pixels are different for different images so normalized deviation would be a better parameter to compare different images.

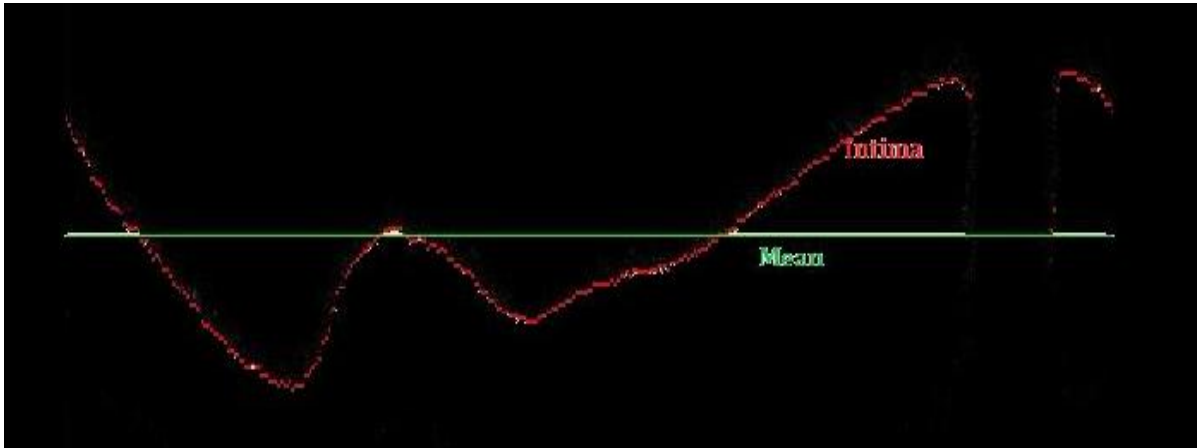
$$(\sigma_{\text{intima}})_N = \frac{\sum_{j=1}^p (X_j - D_c)^2}{p}$$

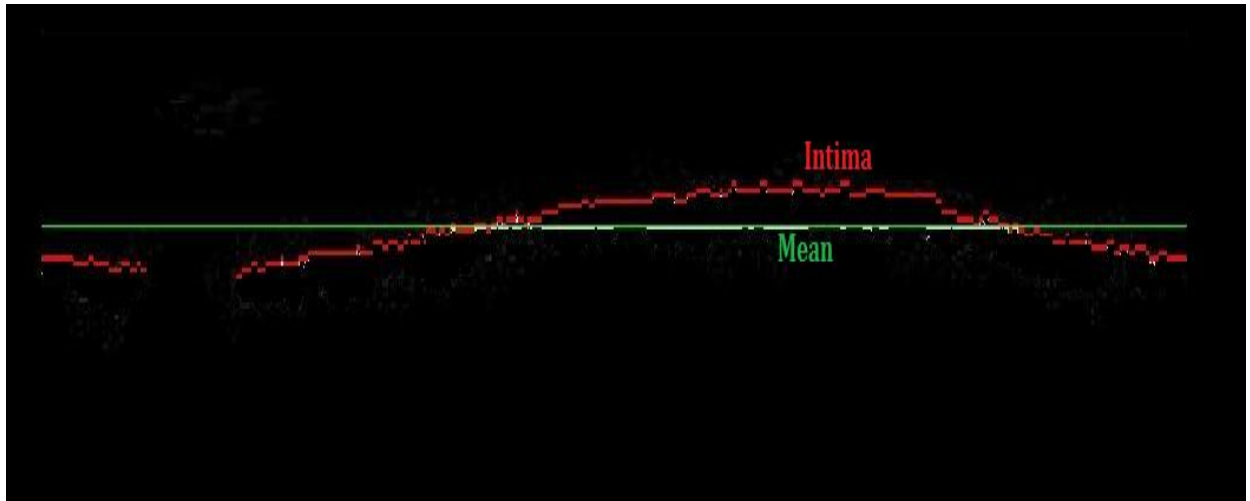
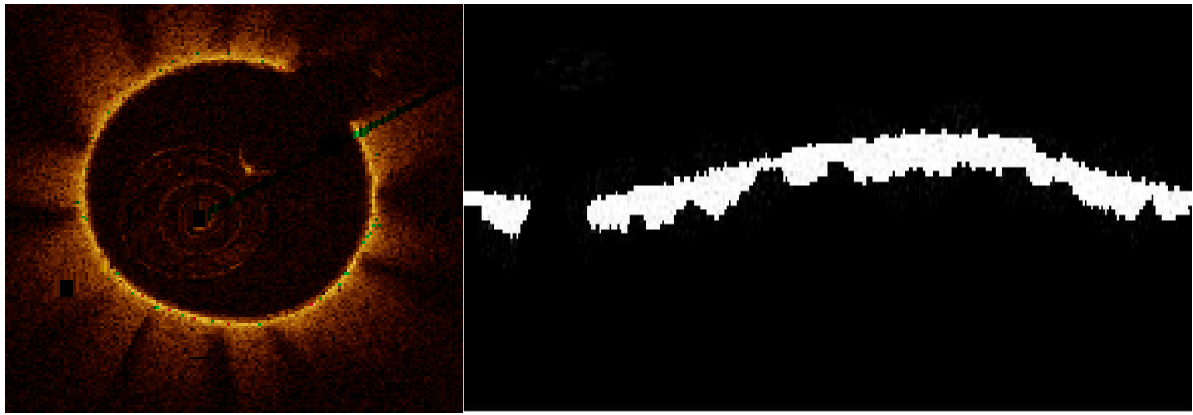


Output of intimal mapping



Planar Image after Conformal Mapping





### 1.3. INFERENCE

<u>Sample</u>	<u>Standard Deviation</u>	<u>Standard deviation(Normalized)</u>
Figure-1	45.2321	0.0742
Figure-2	12.4216	0.0143

Ratio between the normalised standard deviations of two images:

$$(0.0742/0.0143) = 5.88$$



## **CHAPTER-2**

# **SEGMENTATION OF OTHER LAYERS**

## 2.1. GRADIENT APPROACH :

As we know, the gradient is derived from general concept of derivative of a function in 1-D to a function in multi-dimension.

The gradient of a function  $f$  is defined as an unique vector field which is directional derivative of  $f$  along  $v$  at each  $x$ -coordinate calculated by taking its dot product with any vector  $v$ . That is,

$$(\nabla f(x)) \cdot v = D_v f(x)$$

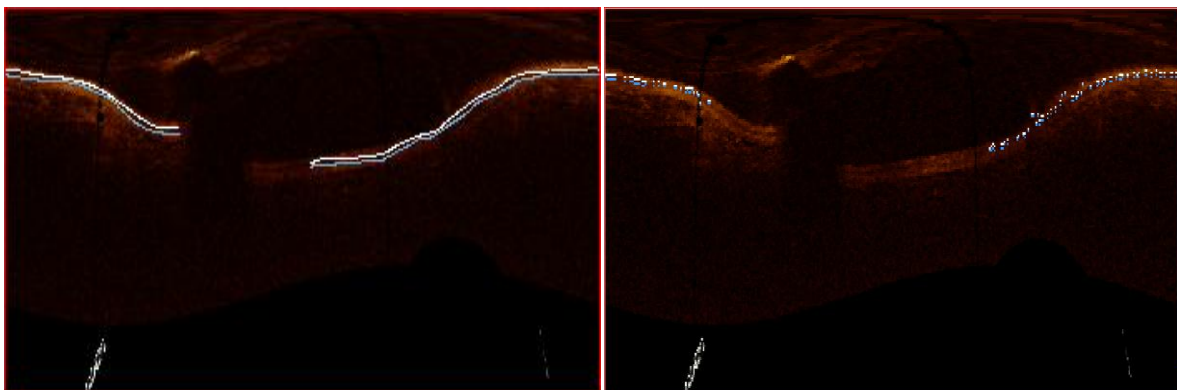
Similarly we can represent gradient in rectangular coordinate system as follows:

$$\nabla f = \frac{\partial f}{\partial x_1} \mathbf{e}_1 + \dots + \frac{\partial f}{\partial x_n} \mathbf{e}_n$$

where the  $\mathbf{e}_i$ s are the orthogonal unit vectors those are pointing in the coordinate axis directions. In the 3-D Cartesian coordinate system, this is given by

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}$$

where  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  are the standard unit vectors.



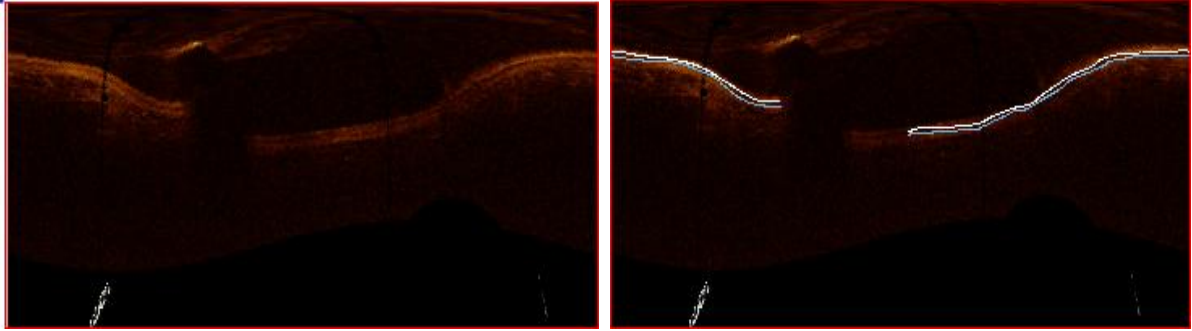
Actual Output

Output of Gradient Method

## **2.2 MANUAL SEGMENTATION :**

Manual segmentation is done taking graphical input for preparation of training set.

Spline interpolation is done to connect the discrete points.



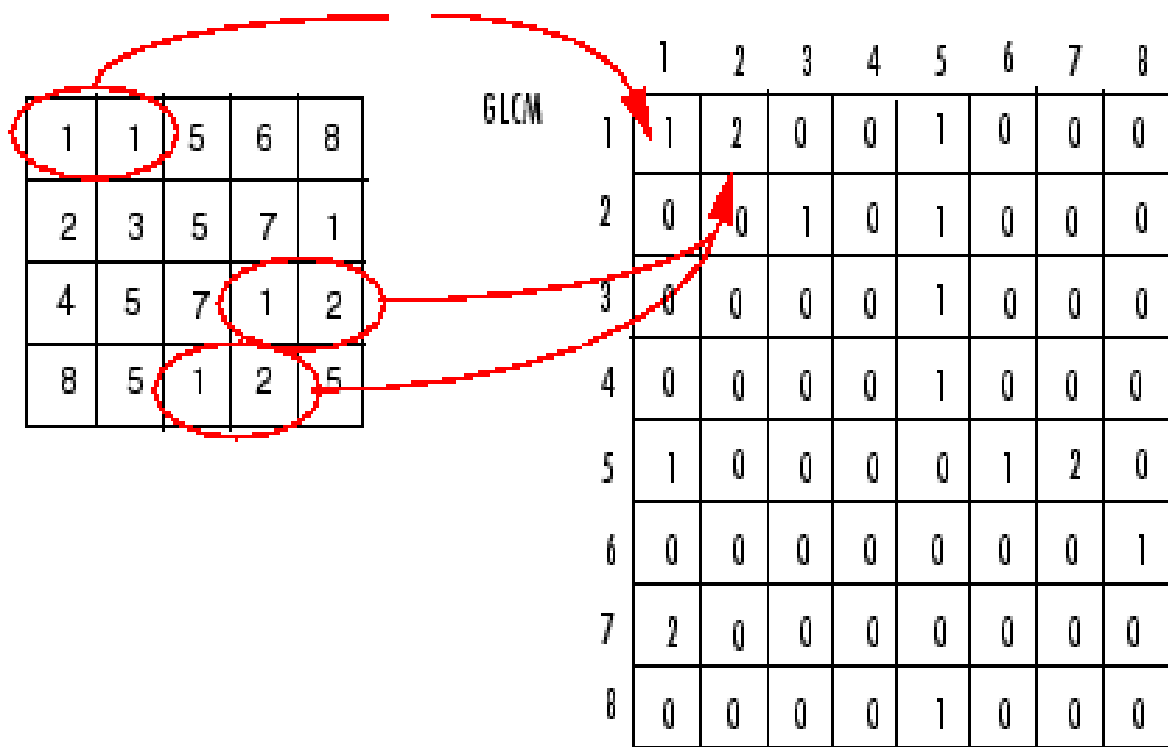
Planar Image before manual segmentation

Planar Image after manual segmentation

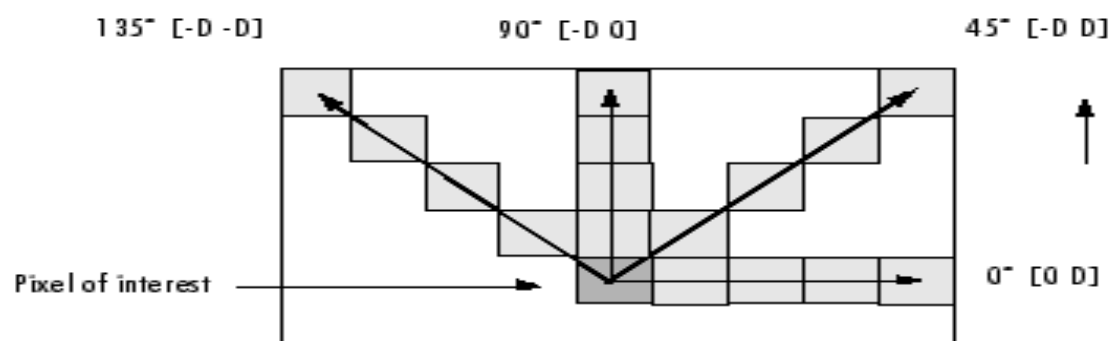
## **2.3. GRAY-LEVEL CO-OCCURRENCE MATRIX:**

A co-occurrence matrix or co-occurrence distribution (less often coöccurrence distribution or coöccurrence matrix) is a distribution or matrix that is defined over an image to be the matrix of co-occurring values at a given offset. A co-occurrence matrix  $C$  can be defined over an  $n \times m$  in an image  $I$ , with an offset of parameter  $(\Delta x, \Delta y)$ , as:

$$C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$



### 2.3.1. OFFSET



For 8x8 neighbourhood offset =

$$[1 -1; 1 0; 1 1; 0 1; -1 1; 0 -1; -1 -1; -1 0]$$

## **2.4. FEATURE EXTRACTION:**

### **CONTRAST-**

Calculates the local transitions in the GLCM.

$$\text{Contrast} = \sum_{i,j} |i - j|^2 p(i, j)$$

### **CORRELATION-**

Calculates the joint probability of occurrence of a specific pixel pairs.

$$\text{Correlation} = \sum_{i,j} \frac{|i - \mu_i| |j - \mu_j|}{\sigma_i \sigma_j} p(i, j)$$

### **ENERGY-**

Results in the sum of squared elements present in the GLCM. Also known as angular second moment or the uniformity in the image.

$$\text{Energy} = \sum_{i,j} p(i, j)^2$$

### **HOMOGENEITY-**

Verifies the closeness of the distribution of elements in the GLCM diagonal to the GLCM matrix.

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i, j)}{1 + |i - j|}$$

## **2.5. SUPPORT VECTOR MACHINE (SVMs)**

In machine learning, SVMs are supervised learning models which includes learning algorithms that analyzes data and recognizes the patterns, used for regression analysis and classification of data. Given a set of training datas, each marked as belonging to one of the two categories, an SVM algorithm builds a model that assigns new datas into one of the two category, so falls in the category of a non-probabilistic binary classifier (linear). An SVM model is representation of the datasets as points in space, mapped so that the datasets of the separate categories are divided by a clear gap i.e; wide margin. New datasets are then mapped into the same space and are predicted to belong to one of the category based on the features that decides which side of the margin they belong to.

### **Support Vector Machine (SVM) classifier**

- Wide margin
- Cost function
- Slack variables
- Loss functions
- Optimization

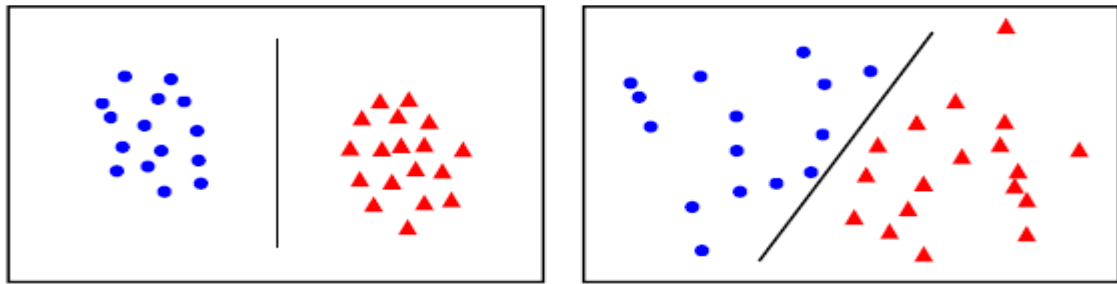
### **Binary Classification**

Given the training data  $(x_j, y_j)$  for  $j = 1 \dots N$ , with  $\mathbf{x}_i \in \mathbf{R}^d$  and  $y_i \in \{-1, 1\}$ , learn a

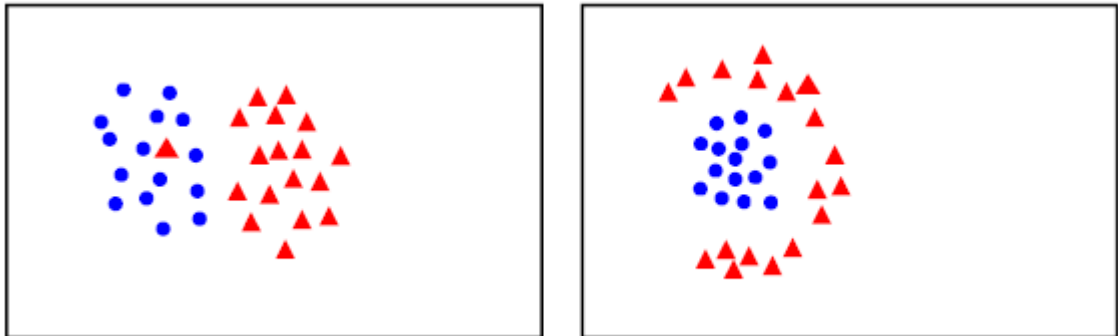
classifier  $f(x)$  such that  $\mathbf{f}(\mathbf{x}_i) = \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$

i.e.  $y_i * f(x_i) > 0$  for a correct classification.

Linearly Separable



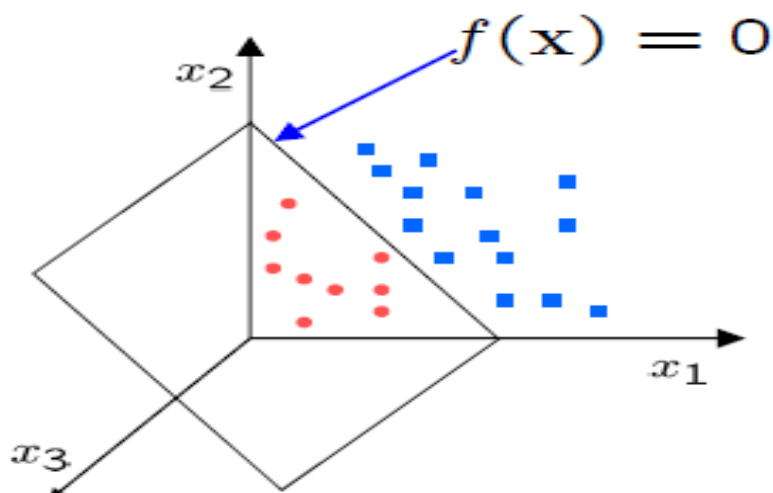
Not Linearly separable



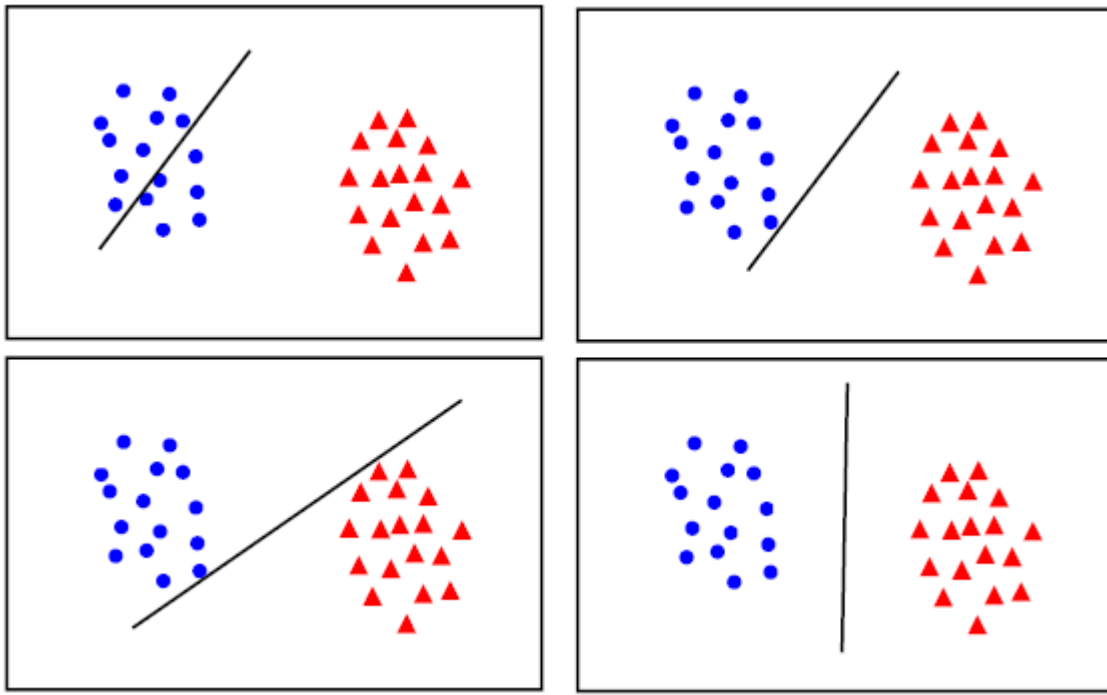
**A linear classifier has the form**

In 3D the discriminant is a plane, and can be expanded to nD where it is a hyperplane For a K-NN classifier it is required to 'carry' the data for training. A linear classifier, is in which the data for training is actually used to learn  $\mathbf{w}$  (margin) and then discarded. For only classifying new data  $\mathbf{w}$  is needed.

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$



FINDING BEST  $\mathbf{w}$ :



maximum margin solution: most stable under worst case of the inputs.

$$\min_{\omega \in \mathbb{R}^d, \xi_i \in \mathbb{R}^+} ||\mathbf{w}'||^2 + C \sum_i^N \xi_i$$

$$\text{subject to} \quad \mathbf{y}_i(\mathbf{w}'^T \mathbf{x} + b) \geq 1 - \xi_i \quad \text{for } i=1, \dots, N$$

If  $\xi_i$  is sufficiently large then every constraint can be satisfied

- Regularization parameter is denoted by C :

- large margin  $\rightarrow$  small C which allows the constraints to be easily ignored

- narrow margin  $\rightarrow$  large C which makes the constraints difficult to ignore

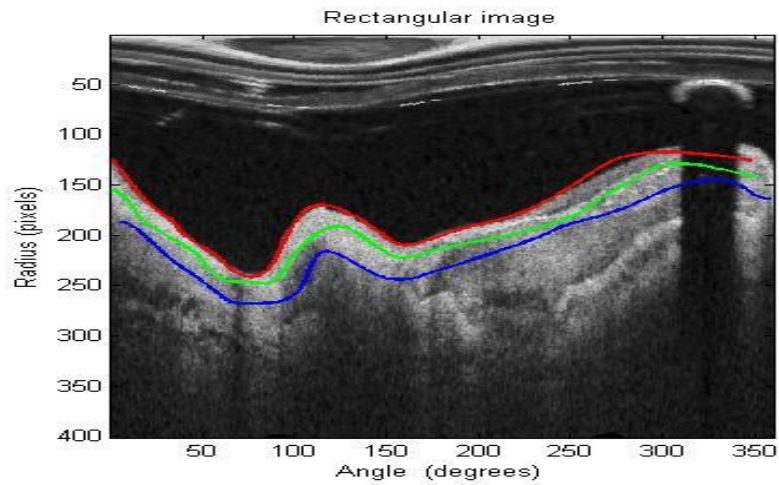
- $C = \infty$  enforces all constraints: hard margin

- As it is a optimization problem of second order and hence has a unique minimum. Note, that there is the only parameter, C.

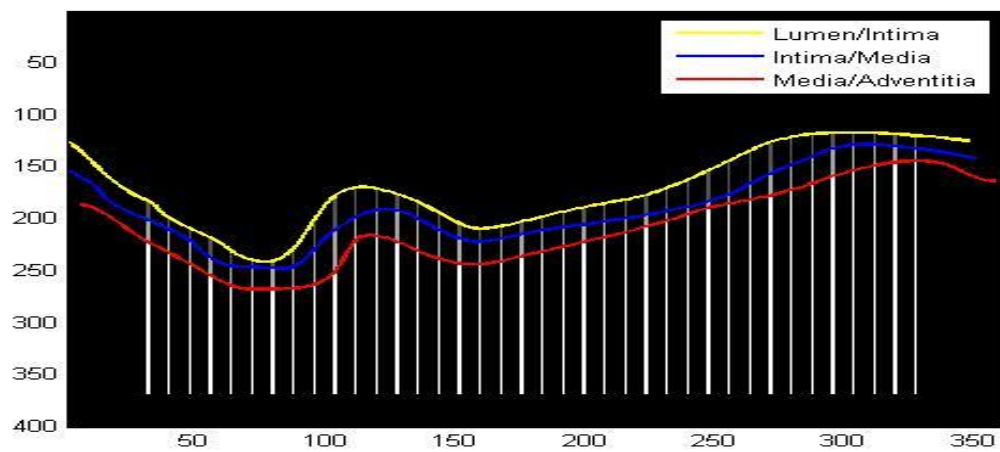


## 2.6 RESULTS

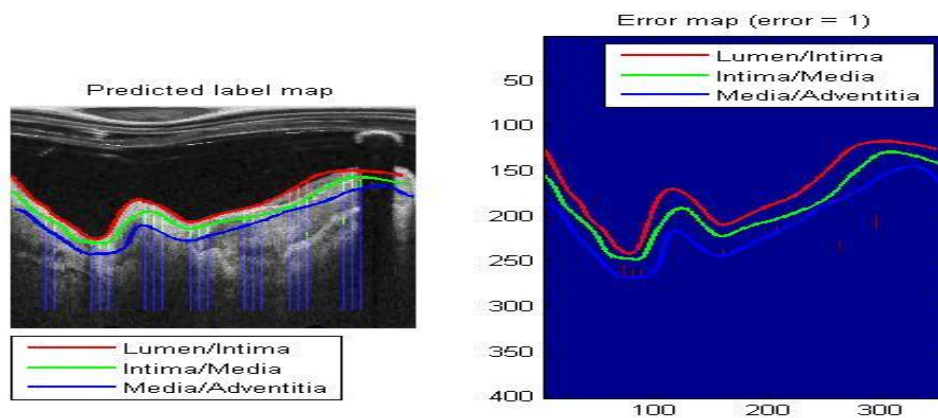
Using the data obtained from the features we trained the SVM and the output is as follows



Result of Implementation of SVM



Level Maps used in SVM



Level Maps with predicted values in SVM

Error Map after SVM

## **CHAPTER-3**

### **POINT MATCHING**

**3.1. Iterative Closest Point (ICP)** is a point matching algorithm which is employed to minimize the difference between two sets of points. In this algorithm, one point set, i.e; the *reference*, or *target set*, is kept fixed, whereas the other set, the *source*, is transformed to best match the reference. The algorithm is an iterative approach to revise the transformation (combination of translation and rotation) which is needed in order to minimize the distance between the source and the reference point cloud.

#### OBJECTIVE:

To find a transformation between 2 set of datas

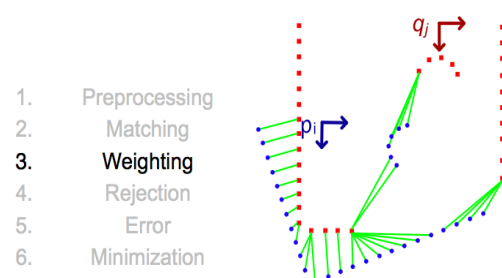
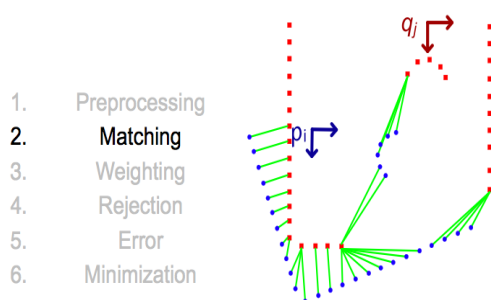
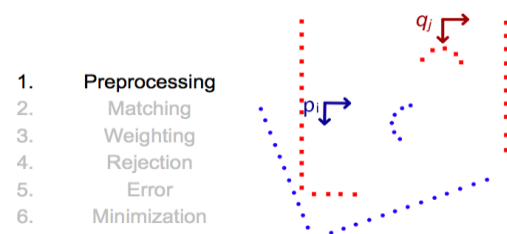
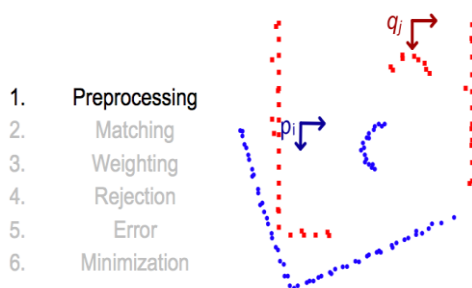
- **INPUT:**

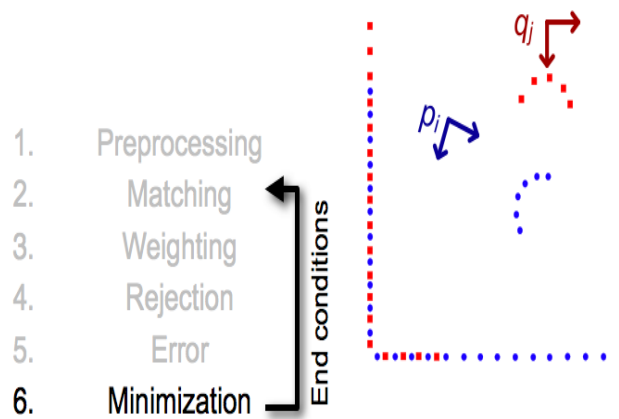
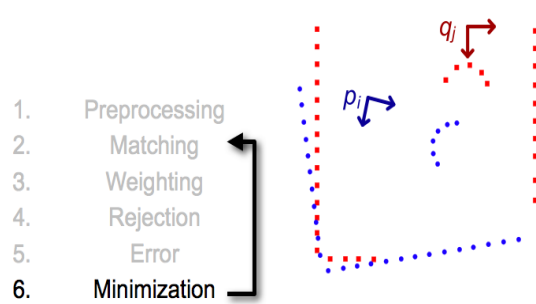
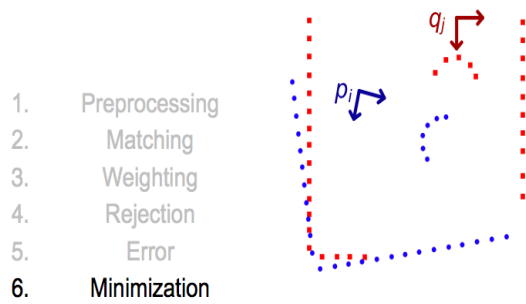
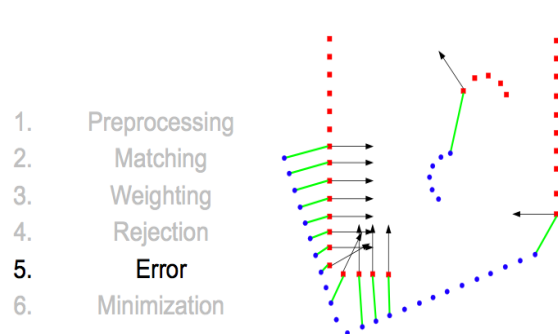
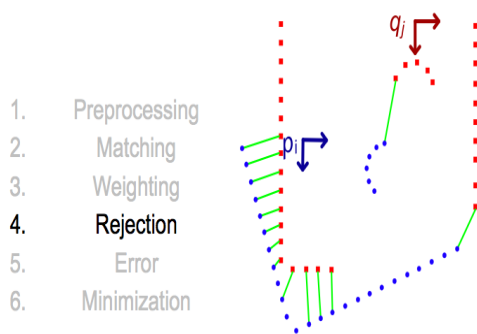
Reference point cloud

Data point cloud

- **OUTPUT:**

Find transformation between data and reference.





### **3.2. ALGORITHM**

#### **1. PRE-PROCESSING:**

clean data

#### **2. MATCHING:**

associate points from data to reference

use neighbour search

can use some features

#### **3. WEIGHTING:**

change priority of some pairs

#### **4. REJECTION:**

remove some pairs

#### **5. ERROR:**

calculate error for each and every pair

#### **6. MINIMIZATION:**

find the best transform

#### **7. LOOP TO 2. UNLESS CONVERGENCE.**

- Objective:

Find point in order to associate to each point in data-set, based on position, also normal vectors, colors ...

- Means:

Additional information obtained as additional dimensions of point , nearest neighbour search.

- Done a lot:

Needs to be efficient.

### **3.3 K-D TREE**

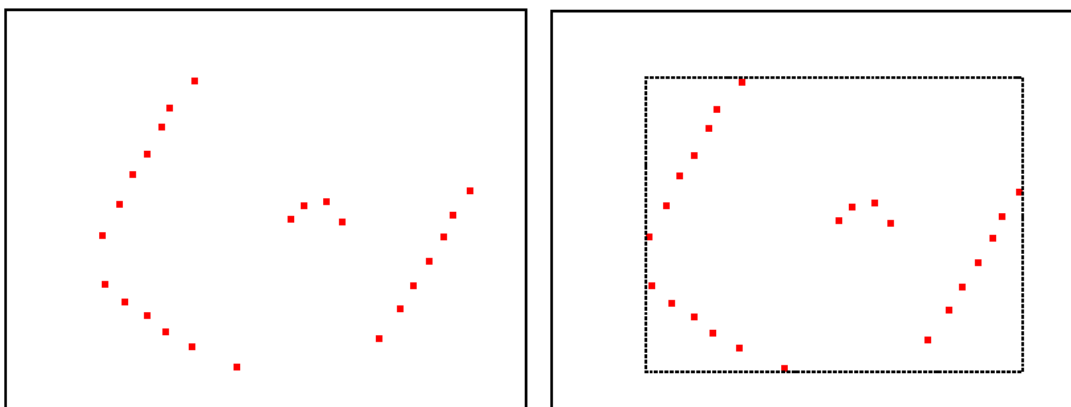
In general, a  $k$ -d tree or *kdimensional tree* is a space-divider data structure used for organizing the points in a  $k$ -D space. These  $Kd$  trees are one of the useful data structure for many applications, for example, searches which involves a multidimensional search key .  $Kd$  trees can be thought of as a special case of binary space partitioning trees.

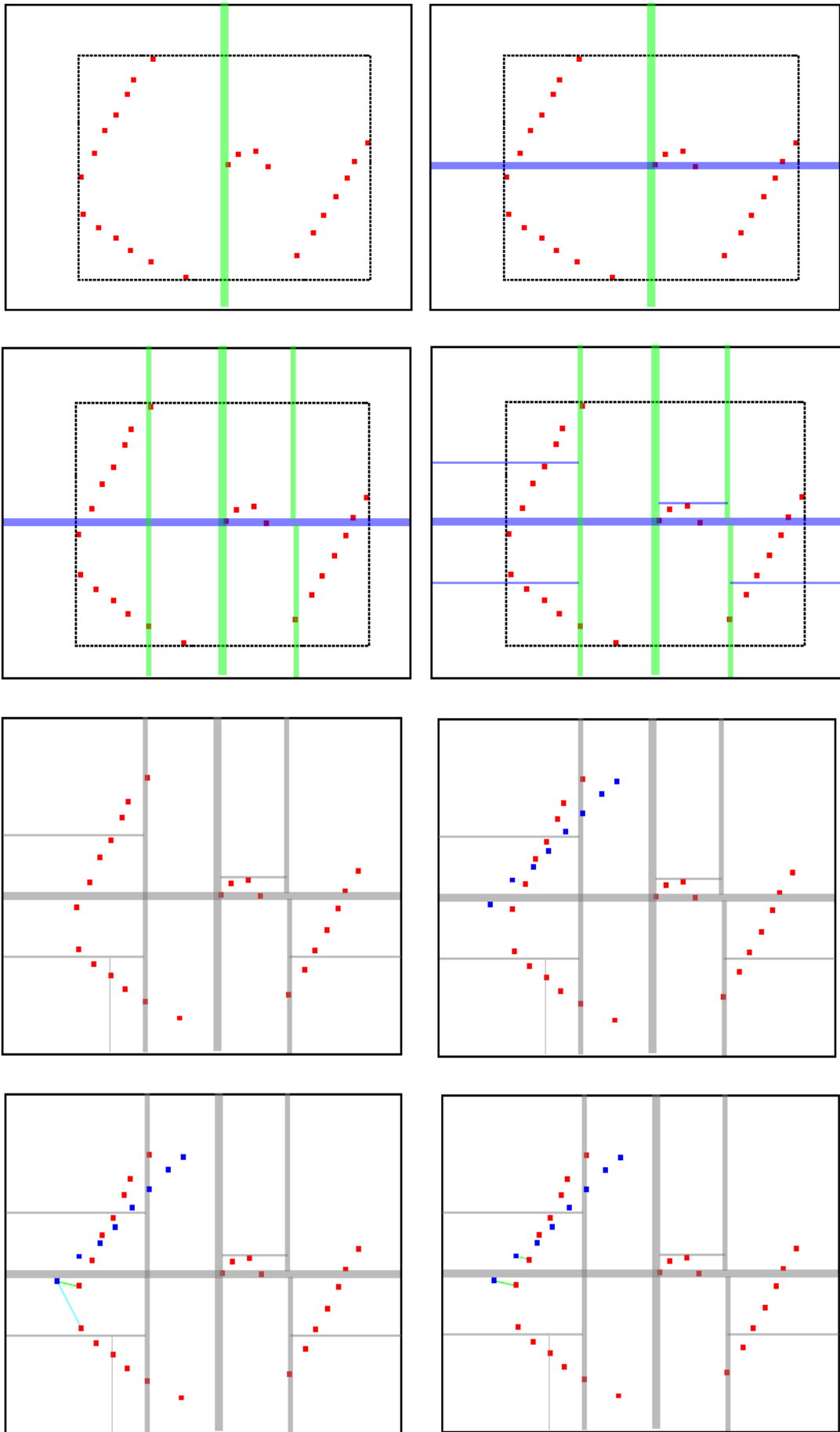
#### **3.3.1. CONSTRUCTION**

Since there exists many ways to choose the axis aligned splitting planes, so there exists many different ways which helps to construct  $kd$  trees. The canonical method to construct  $kd$  tree has the following constraints:

1. As one descend down the tree, by one cycles through the axes which is used to select the splitting planes. (For e.g; in a 3-D tree, the root is having an  $x$ -aligned plane, then the root's both of the children will have  $y$ -aligned planes, so the root's grand-children all will have  $z$ -aligned planes, then the root's great-grandchildren will all have  $x$ -aligned planes, similarly the root's great-great-grandchildren will all have  $y$ -coordinate aligned planes, and so on.)
2. By selecting the median of the points are being inserted into the subtree, with respect to the coordinates in the axis which is being used to create splitting planes.

Build the tree for reference :





### **3.3.2. NEAREST NEIGHBOUR SEARCH**

The NN(nearest neighbour search) algorithm which aims to find the point in the tree which is nearest to a given point as the input. This search can be efficiently done by using properties of the tree to quickly eradicate large proportions of the search space.

Searching for the nearest neighbour in a  $k$ -dimension tree proceeds as follows:

1. This algorithm is a recursive approach moves down the tree beginning from the root node i.e. it goes right or left depending upon whether the point is greater or lesser than the present or current node in this split dimension.
2. Once this algorithm reaches any leaf node, then it saves the same node as the "present best"
3. As always in the recursion the algorithm unwinds the tree, by performing the following steps at every node:
  1. If the present node is closer than present best, then it becomes the present best.
  2. This algorithm also checks whether there may exist some points on the other side of the dividing plane which are closer to the search point than the present best. This is done by intersection of the hypersphere with a dividing hyperplane around the search point which has a radius same as the present nearest distance.
    1. If the hypersphere crosses the plane, then there is a possibility of getting nearer points on the other portion of the plane, so this algorithm must move down to the other branches of the tree from the present node in order to look for closer points, by following the same process(recursive) as the entire search.



2. If this hypersphere doesn't intersect with the dividing plane, then this algorithm continues by walking up the tree, and the whole branch on the other half of that node is eliminated.

4. When the algorithm completes the same process for the root node, then we can say the search is complete.

### 3.3.3. COMPLEXITY OF K-D TREE-

	AVERAGE CASE	WORST CASE
Space	$O(n)$	$O(n)$
Search	$O(\log(n))$	$O(n)$
Insert	$O(\log(n))$	$O(n)$
Delete	$O(\log(n))$	$O(n)$

### 3.4. TRANSFORMATIONS

$$\mathcal{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(rx) & -\sin(rx) \\ 0 & \sin(rx) & \cos(rx) \end{bmatrix}$$

$$\mathcal{R}_y = \begin{bmatrix} \cos(ry) & 0 & \sin(ry) \\ 0 & 1 & 0 \\ -\sin(ry) & 0 & \cos(ry) \end{bmatrix}$$

$$\mathcal{R}_z = \begin{bmatrix} \cos(rz) & -\sin(rz) & 0 \\ \sin(rz) & \cos(rz) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{T}_{xyz} = [T_x \quad T_y \quad T_z] \quad \text{where } \mathcal{T}_{xyz} \text{ is translation matrix}$$

$$\mathcal{R}_{xyz} = \mathcal{R}_x * \mathcal{R}_y * \mathcal{R}_z \quad \text{where } \mathcal{R}_{xyz} \text{ is total rotation matrix}$$

Let data set be  $\mathcal{P}$  and reference data set  $\mathcal{M}$  Transformed Matrix is  $\mathcal{T}$

$$\mathcal{T} = \mathcal{R}_{xyz} * \mathcal{D} + \mathcal{T}_{xyz}$$

### 3.5. ERROR MINIMIZATION-

Point to Point

$$d_{RMS} = \mathcal{E}(\mathcal{R}_{xyz}, \mathcal{T}_{xyz}) = \sum_i = I^N \|\mathcal{R}_{xyz} \mathcal{P}_i + \mathcal{T}_{xyz} - \mathcal{M}_i\|^2$$

Where  $\mathcal{M}_i$  is the reference point corresponding to data point  $\mathcal{P}_i$ .

Point to Plane

$$d_{RMS} = \mathcal{E}(\mathcal{R}_{xyz}, \mathcal{T}_{xyz}) = \sum_i = I^N \|\mathcal{R}_{xyz} \mathcal{P}_i + \mathcal{T}_{xyz} - \mathcal{M}_i\|^2$$

Where  $\mathcal{M}_i$  is the projection of  $\mathcal{P}_i$  on the surface.

Getting the best transformation:

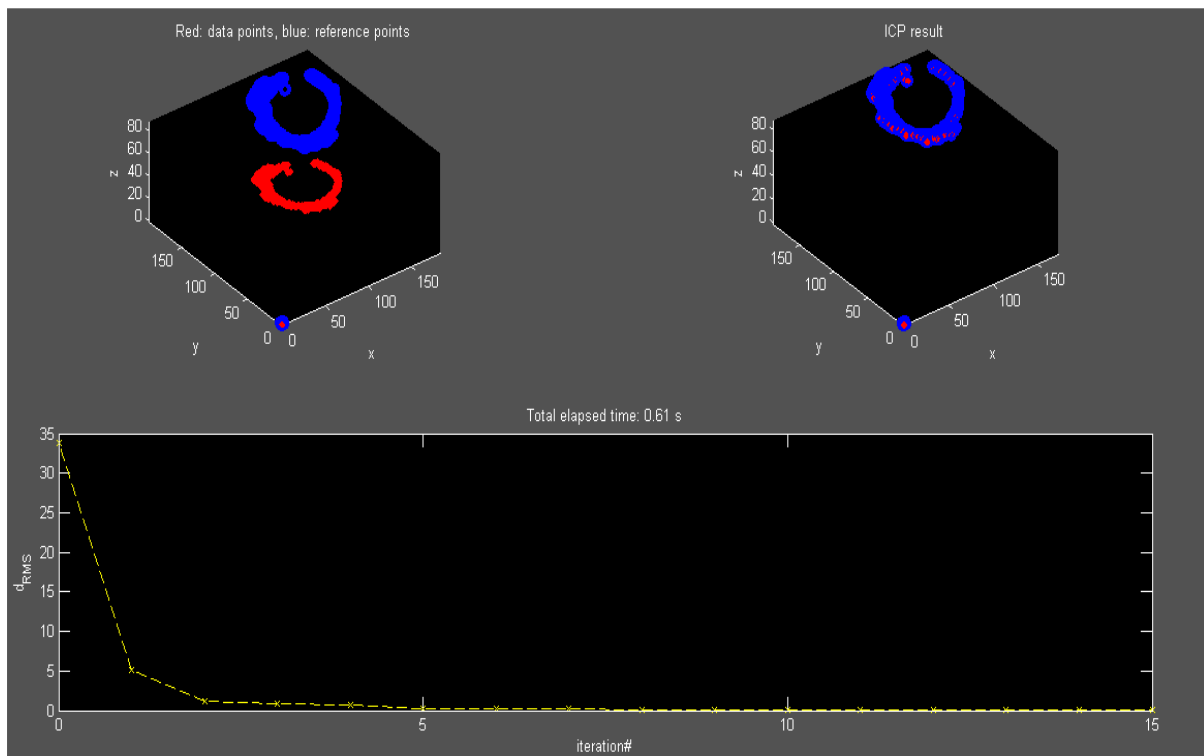
$$(\mathcal{R}_{xyz}, \mathcal{T}_{xyz}) = \underset{\mathcal{R}_{xyz}, \mathcal{T}_{xyz}}{\operatorname{argmin}} \mathcal{E}(\mathcal{R}_{xyz}, \mathcal{T}_{xyz})$$

Iterative Process :

$$(\mathcal{R}_{xyz}, \mathcal{T}_{xyz})_{\hat{k}} = \underset{\mathcal{R}_{xyz}, \mathcal{T}_{xyz}}{\operatorname{argmin}} \sum_i = I^N \|\mathcal{R}_{xyz} p_i^{\hat{k}} + \mathcal{T}_{xyz} - \mathcal{M}_i^{\hat{k}}\|^2$$

$$\text{Where } p_i^{\hat{k}} = \mathcal{R}_{xyz}^{\hat{k}-1} p_i^{\hat{k}-1} + \mathcal{T}_{xyz}^{\hat{k}-1}$$

### 3.6. RESULT OF POINT MATCHING USING ICP-



Number of Iterations-15

Time Elapsed =0.61seconds

k-d tree with k=3

Final Error=0

## **CONCLUSION:**

The algorithms mentioned above was done using MATLAB 2013a. These algorithms are efficient enough to segment the walls of inter-coronary artery images. More larger the dataset used for training SVM better is the result obtained. The algorithm used for 3-d construction of artery using some biasing reduces computational complexity to a greater extent. Finally the results obtained are validated for around 1000 samples.

## **REFERENCES:**

- [1] Tony F. Chan, *Member, IEEE*, and Luminita A. Vese, "Active Contours Without Edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp, Feb. 2001.
- [2] Gozde Unal, *Senior Member, IEEE*, Susann Bucher, Stephane Carlier, Greg Slabaugh, Tong Fang, and Kaoru Tanaka," Shape-Driven Segmentation of the Arterial Wall in Intravascular Ultrasound Images", *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, VOL. 12, NO. 3, MAY 2008.
- [3] J. Klingensmith, R. Shekhar, and D. Vince, "Evaluation of threedimensional segmentation algorithms for the identification of luminal and medial-adventitial borders in intravascular ultrasound images," *IEEE Trans. Med. Imag.*, vol. 19, no. 10, pp. 996–1011, Oct. 2000.
- [4] Rosenberg, J. B. (1985). "Geographical Data Structures Compared: A Study of Data Structures Supporting Region Queries". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*.
- [5] Cristianini, Nello; and ShaweTaylor, John; *An Introduction to Support Vector Machines and other kernel based learning methods*, Cambridge University Press, 2000. ISBN 0521780195 ([1](<http://www.supportvector.net>) SVM Book)